package com.twitter.product\_mixer.component\_library.scorer.tensorbuilder

import com.twitter.ml.api.thriftscala.FloatTensor

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.feature.FeatureWithDefaultOnFailure

import com.twitter.product\_mixer.core.feature.ModelFeatureName

import com.twitter.product\_mixer.core.feature.featuremap.featurestorev1.FeatureStoreV1FeatureMap.\_

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.feature.featurestorev1.FeatureStoreV1CandidateFeature

import com.twitter.product\_mixer.core.feature.featurestorev1.FeatureStoreV1QueryFeature

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import inference.GrpcService.ModelInferRequest.InferInputTensor

class QueryInferInputTensorBuilder[-Query <: PipelineQuery, +Value](

builder: InferInputTensorBuilder[Value],

features: Set[\_ <: Feature[Query, \_] with ModelFeatureName]) {

def apply(query: Query): Seq[InferInputTensor] = {

val featureMap = query.features.getOrElse(FeatureMap.empty)

features.flatMap { feature =>

val queryFeatureValue: Value = feature match {

case feature: FeatureStoreV1QueryFeature[Query, \_, Value] =>

featureMap.getFeatureStoreV1QueryFeature(feature)

case feature: FeatureStoreV1CandidateFeature[Query, \_, \_, Value] =>

throw new UnexpectedFeatureTypeException(feature)

case feature: FeatureWithDefaultOnFailure[Query, Value] =>

featureMap.getTry(feature).toOption.getOrElse(feature.defaultValue)

case feature: Feature[Query, Value] =>

featureMap.get(feature)

}

builder.apply(feature.featureName, Seq(queryFeatureValue))

}.toSeq

}

}

case class QueryBooleanInferInputTensorBuilder[-Query <: PipelineQuery](

features: Set[\_ <: Feature[Query, Boolean] with ModelFeatureName])

extends QueryInferInputTensorBuilder[Query, Boolean](BooleanInferInputTensorBuilder, features)

case class QueryBytesInferInputTensorBuilder[-Query <: PipelineQuery](

features: Set[\_ <: Feature[Query, String] with ModelFeatureName])

extends QueryInferInputTensorBuilder[Query, String](BytesInferInputTensorBuilder, features)

case class QueryFloat32InferInputTensorBuilder[-Query <: PipelineQuery](

features: Set[\_ <: Feature[Query, \_ <: AnyVal] with ModelFeatureName])

extends QueryInferInputTensorBuilder[Query, AnyVal](Float32InferInputTensorBuilder, features)

case class QueryFloatTensorInferInputTensorBuilder[-Query <: PipelineQuery](

features: Set[\_ <: Feature[Query, FloatTensor] with ModelFeatureName])

extends QueryInferInputTensorBuilder[Query, FloatTensor](

FloatTensorInferInputTensorBuilder,

features)

case class QueryInt64InferInputTensorBuilder[-Query <: PipelineQuery](

features: Set[\_ <: Feature[Query, \_ <: AnyVal] with ModelFeatureName])

extends QueryInferInputTensorBuilder[Query, AnyVal](Int64InferInputTensorBuilder, features)

case class QuerySparseMapInferInputTensorBuilder[-Query <: PipelineQuery](

features: Set[\_ <: Feature[Query, Option[Map[Int, Double]]] with ModelFeatureName])

extends QueryInferInputTensorBuilder[Query, Option[Map[Int, Double]]](

SparseMapInferInputTensorBuilder,

features)